

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

PERSONALWEB TECHNOLOGIES, LLC.
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

v.

NEC CORPORATION OF AMERICA, INC.,

Defendant.

§
§
§
§
§
§
§
§
§
§
§

Civil Action No. 6:11-cv-00655 (LED)

JURY TRIAL DEMANDED

PERSONALWEB TECHNOLOGIES, LLC.
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

v.

GOOGLE INC. AND YOUTUBE, LLC.

Defendants.

§
§
§
§
§
§
§
§
§
§
§

Civil Action No. 6:11-cv-00656 (LED)

JURY TRIAL DEMANDED

PERSONALWEB TECHNOLOGIES, LLC.
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

v.

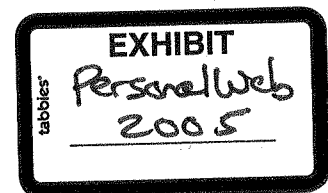
NETAPP, INC.

Defendant.

§
§
§
§
§
§
§
§
§
§
§

Civil Action No. 6:11-cv-00657 (LED)

JURY TRIAL DEMANDED



PERSONALWEB TECHNOLOGIES, LLC.	§	
AND LEVEL 3 COMMUNICATIONS, LLC,	§	
	§	
Plaintiffs,	§	
	§	
v.	§	Civil Action No. 6:11-cv-00658 (LED)
	§	
AMAZON.COM, INC.; AMAZON WEB	§	JURY TRIAL DEMANDED
SERVICES LLC; AND DROPBOX, INC.,	§	
	§	
Defendants.	§	

PERSONALWEB TECHNOLOGIES, LLC.	§	
AND LEVEL 3 COMMUNICATIONS, LLC,	§	
	§	
Plaintiffs,	§	
	§	
v.	§	Civil Action No. 6:11-cv-00660 (LED)
	§	
EMC CORPORATION AND VMWARE,	§	JURY TRIAL DEMANDED
INC.,	§	
	§	
Defendants.	§	

PERSONALWEB TECHNOLOGIES, LLC.	§	
AND LEVEL 3 COMMUNICATIONS, LLC,	§	
	§	
Plaintiffs,	§	
	§	
v.	§	Civil Action No. 6:11-cv-00661 (LED)
	§	
AUTONOMY, INC., HEWLETT-PACKARD	§	JURY TRIAL DEMANDED
COMPANY, AND HP ENTERPRISE	§	
SERVICES, LLC,	§	
	§	
Defendants.	§	

PERSONALWEB TECHNOLOGIES, LLC
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs.

VS.

YAHOO! INC.

Defendant.

§ § § § § § § § § §

Civil Action No. 6:12-CV-00658-LED

JURY TRIAL DEMANDED

PERSONALWEB TECHNOLOGIES, LLC
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

VS.

APPLE INC.

Defendant.

§ § § § §

Civil Action No. 6:12-CV-00660-LED

JURY TRIAL DEMANDED

PERSONALWEB TECHNOLOGIES, LLC
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

VS.

FACEBOOK INC.

Defendant.

§ § § § § § § § § §

Civil Action No. 6:12-CV-00662-LED

JURY TRIAL DEMANDED

PERSONALWEB TECHNOLOGIES, LLC
AND LEVEL 3 COMMUNICATIONS, LLC,

Plaintiffs,

VS.

MICROSOFT CORPORATION

Defendant.

§ § § § §

Civil Action No. 6:12-CV-00663-LED

JURY TRIAL DEMANDED

PersonalWeb Technologies, LLC's Opening Claim Construction Brief

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	OVERVIEW OF THE TRUE NAME PATENTS.....	1
III.	CLAIM CONSTRUCTION ANALYSIS	2
A.	Claim Construction Analysis of Terms Containing the Word “Identifier.”	2
1.	“Identifier” should be given its plain and ordinary meaning and construed as “something that identifies” or “a value that identifies.”	3
a)	The term “identifier” does not require construction.	3
b)	Defendants’ proposed construction for “identifier” contradicts the claim language.	4
2.	“Substantially Unique Identifier” should be construed as “an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm.”.....	5
3.	“Digital identifier” should be given its plain and ordinary meaning and construed as “an identifier in digital form.”.....	8
4.	“Data item identifier” should be given its plain and ordinary meaning and construed to mean “an identifier for a data item.”	10
5.	“Data identifier” should be construed the same as “substantially unique identifier.”	11
B.	Claim Construction Analysis for Remaining Terms.....	12
1.	“True Name” should be construed as “the substantially unique identifier for a particular data item, calculated in accordance with the description at ’791 12:54 to 13:9.”	12
a)	The specification provides a clear description of how to calculate a True Name.	13
b)	Defendants’ proposed construction improperly limits the claims to a preferred embodiment.....	13
2.	“Data item(s)” should be construed as “sequence of bits.”	14
a)	The specification defines “data item” as “sequence of bits.”	14
b)	Defendants’ proposed constructions are wholly unsupported by the intrinsic evidence.	15

c)	“Data item” does not render claim 30 ambiguous.	16
3.	“Data file(s)” should be construed as “a named data item having one or more data segments.”	17
a)	PersonalWeb’s construction looks to the guidance provided in the specification.	17
b)	Defendants’ proposed constructions import extraneous limitations into the term “data file.”	18
4.	“File system” should have its ordinary meaning.	18
a)	This term does not require construction.....	18
b)	Defendants’ proposed constructions run afoul of the claim language.	19
5.	“Sufficient number of copies” should be given its plain and ordinary meaning, or should be construed to mean “application-specific criteria regarding the number of copies required.”	19
6.	“Licensed/unlicensed” should be given its plain and ordinary meaning, or should be construed to mean “with license/without license.”	20
7.	“Distributing a set of data files across a network of servers” should be given its plain and ordinary meaning, or should be construed to mean “storing a set of data files on various servers in a network.”	21
C.	Construction Analysis for Means Plus Function Limitations.	22
1.	“Access means for accessing a particular data item using the identifier of the data item.”	22
2.	“Data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item.”	23
3.	“Existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.”	25
4.	“Identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data on the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.”	26

5.	“Local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item.”	28
IV.	CONCLUSION.....	29

TABLE OF AUTHORITIES

Other Authorities

<i>ACTV, Inc. v. Walt Disney Co.</i> , 346 F.3d 1082 (Fed. Cir. 2003)	26
<i>AFG Indus., Inc. v. Cardinal IG Co.</i> , 239 F.3d 1239 (Fed. Cir. 2001)	4
<i>Altiris Inc. v. Symantec Corp.</i> , 318 F.3d 1363 (Fed Cir. 2003)	3, 18
<i>Bell Atl. Network Servs., Inc. v. Covad Commc'ns Grp., Inc.</i> , 262 F.3d 1258 (Fed. Cir. 2001)	5, 9, 11, 12
<i>Callicrate v. Wadsworth Mfg.</i> , 427 F.3d 1361 (Fed. Cir. 2005)	passim
<i>Datamize, LLC v. Plumtree Software, Inc.</i> , 417 F.3d 1342 (Fed. Cir. 2005)	7
<i>Enzo Biochem, Inc. v. Applera Corp.</i> , 599 F.3d 1325 (Fed. Cir. 2010)	16
<i>Interactive Gift Exp., Inc. v. Compuserve Inc.</i> , 256 F.3d 1323 (Fed. Cir. 2001)	9
<i>Level 3 Commc'ns, LLC v. Limelight Networks, Inc.</i> , 589 F. Supp. 2d 664 (E.D. Va. 2008)	4
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed Cir. 2005)	passim
<i>Versa Corp. v. Ag-Bag International Ltd.</i> , 392 F.3d 1325 (Fed. Cir. 2004)	passim
<i>Wenger Mfg. Inc. v. Coating Mach. Sys., Inc.</i> , 239 F.3d 1225 (Fed. Cir. 2001)	passim
<i>Young v. Lumenis, Inc.</i> , 492 F.3d 1336 (Fed. Cir. 2007)	7, 16, 25

Regulations

35 U.S.C. § 112.....	passim
----------------------	--------

I. INTRODUCTION

PersonalWeb's asserted patents¹ (the "True Name" patents) claim inventions directed to various methods of identifying data (such as files or portions of files) to improve the speed and efficiency of accessing data, increase effective storage capacity of data, and reduce bandwidth needs in transferring data.

II. OVERVIEW OF THE TRUE NAME PATENTS

On April 11, 1995, inventors David Farber and Ronald Lachman filed a patent application directed toward the use of "substantially unique identifiers" to identify and thereafter to track, manage, store, and account for data. Farber and Lachman foresaw the then-coming world of rapidly expanding and globally distributed data and networks, and realized that it would be important to have more efficient and reliable ways of identifying data for accessing, storing, and transferring data. Their solution was elegant: identify data in the system in a manner that is independent of subjective names (such as user-given file names, file locations, or other metadata) and instead is based on the data itself. That way, data could be identified across multiple, complex networks and file systems, independent of different naming practices or different languages, and independent of the different ways people may choose to name, identify, or store their data. Farber and Lachman developed and claimed various techniques for using such content-based identifiers to rapidly access and efficiently store and transfer data.

As explained in the patents-in-suit, content-based identifiers are generated by applying a function, such as a message digest or hash function, to data in order to calculate a value that identifies that data for specific purposes within the system. For example, these values may be

¹ The asserted patents include U.S. Patent Nos. 5,978,791, 6,415,280, 6,928,442, 7,802,310, 7,945,539, 7,945,544, 7,949,662, 8,001,096, and 8,099,420, attached as Exhibits 3–11.

used as a key that maps to storage locations of the corresponding data so that the data can be accessed for such purposes as being retrieved, deleted, or de-duplicated. As another example, these values may be used to determine whether particular data is present at a given location to avoid transferring it to that location if already there.

The patents also teach that a set of data may be divided into parts, each part having its own identity, so that the data can be accessed via the identities of its respective parts. Using this invention, data no longer needs to be stored or transmitted monolithically. Identical sets of data (or parts thereof) need not be stored or transmitted more than needed.

III. CLAIM CONSTRUCTION ANALYSIS²

A. Claim Construction Analysis of Terms Containing the Word “Identifier.”

A central dispute between the parties is over the construction of various multi-word terms containing the word “identifier.” Ignoring the actual claim language and specifications, Defendants collectively assert that any term with the word “identifier,” for example “digital identifier” and “data item identifier,” must uniformly be construed to mean what Defendants contend is the proper construction of a much narrower term—“substantially unique identifier.” But the claim language and patent specification describing each of these terms plainly communicates a different meaning for each term. Accordingly, for the terms in which “identifier” appears, no construction is necessary. To the extent construction is necessary, it must follow either the express language of the claims or the express definition in the specification, which in each instance provides sufficient explanation or qualification as to what each “identifier” refers.

² The terms for which the parties have reached agreed constructions are attached as Exhibit 1. The disputed terms, with the parties’ proposed constructions, are attached as Exhibit 2.

1. “Identifier” should be given its plain and ordinary meaning and construed as “something that identifies” or “a value that identifies.”

The term “identifier” is merely a component of the many “identifier” terms. As certain Defendants agree, the plain and ordinary meaning should govern and the term “identifier” must be construed as “something that identifies.” *See* Dkt. No. 157-2 at 13-14.³ Google, however, insists that the Court disregard this term’s plain and ordinary meaning and instead impose on it a limitation contained in an entirely separate term. In particular, Defendants argue that “identifier” must take on the limitations of “substantially unique identifier,” which, as explained *infra* in Section A.3, is generated by applying a function to all of the data in the data item and only the data in the data item. *Id.* at 14 (emphasis added). Google thus seeks to import specific constraints from a specific term and make those a part of the broader “identifier” definition. Such a construction impermissibly changes and even contradicts the meaning of many claims that contain the term “identifier.”

a) The term “identifier” does not require construction.

The individual word “identifier,” which appears throughout many of the claims, has a readily understood plain and ordinary meaning that requires no separate construction. *See e.g.*, ’791 at claims 1, 2, 3, 4, 30, 31, 32, 33, 35, 41; ’310 at claims 1, 5, 6, 70, 81, 86; ’539 at claims 10, 21, 34; ’662 at claims 1, 3, 20; ’096 at claims 1, 2, 81, 83. Simply put, an “identifier” is “something that identifies.” That plain and ordinary meaning could alternatively be expressed as “a value that identifies,” which means the same thing as “something that identifies.” *See Altiris Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed Cir. 2003) (holding there is a “heavy

³ Docket number references throughout this brief refer to documents filed in the case styled *PersonalWeb Techs., LLC and Level 3 Commc’ns., LLC v. Google, Inc. and YouTube, LLC*, CA No. 6:11-cv-655.

presumption” that claim terms carry their ordinary and customary meaning). Since this Court’s claim constructions will become jury instructions, *AFG Indus., Inc. v. Cardinal IG Co.*, 239 F.3d 1239, 1247 (Fed. Cir. 2001), the phrase “something that identifies” is preferred to the alternative expression, “a value that identifies,” because it is more jury friendly.

Indeed, several of the Defendants either have not asked the Court to construe this term, or have agreed that the plain and ordinary meaning, or some close alternative, should apply. *See* Dkt. No. 157-2 at 13-14 (for example, NECAM proposes that the Court adopt a “[p]lain and ordinary meaning”). This supports PersonalWeb’s position. *See Level 3 Commc’ns, LLC v. Limelight Networks, Inc.*, 589 F. Supp. 2d 664, 689 (E.D. Va. 2008) (“The fact that Plaintiff’s proposed construction here was agreed upon by parties to an unrelated litigation that are non-parties here is useful, but certainly not dispositive.” (footnote omitted)).

b) Defendants’ proposed construction for “identifier” contradicts the claim language.

The proposal of several defendants to construe “identifier” coextensively with “substantially unique identifier” conflicts with both the claim language and the specification. Unambiguous claim language in a number of patents-in-suit expressly uses the term “identifier” in a way that clearly differs from the meaning of “substantially unique identifier.”

Consider, for example, claim 6 of the ’310 patent:

A method as recited in claim 1 wherein the plurality of identifiers in the database are identifiers of licensed content items, and wherein the identifier of each licensed content item is based at least in part on the function of at least some of the data comprising the licensed content item.

’310 at claim 6 (emphasis added). The words “at least some of the data” demonstrate that an “identifier” can be based on less than all of the data comprising the licensed content item.

Additionally, the words “at least in part” show that an “identifier” can be based on data other than the data comprising the licensed content item.

Google’s position, for example, that all “identifiers” must be generated by—as is the procedure for generating a “substantially unique identifier”—“processing all of the data in the data item and only the data in the data item,” directly conflicts with this claim language. Dkt. No. 157-2 at 7-8 (emphasis added). Thus, the ordinary meaning of “identifier” should apply, and this Court should not adopt a construction that is contrary to the unambiguous language contained in many of the claims. *See Bell Atl. Network Servs., Inc. v. Covad Commc’ns Grp., Inc.*, 262 F.3d 1258, 1268 (Fed. Cir. 2001) (holding that there is generally a “heavy presumption” in favor of the ordinary meaning of claim language as understood by one of ordinary skill in the art (internal citations omitted)).

2. “Substantially Unique Identifier” should be construed as “an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm.”

Unlike the commonly used term “identifier,” the phrase “substantially unique identifier” as used in claims 1, 30, 33, and 35 of the ’791 patent does not have an ordinary meaning. Rather, the claims define “substantially unique identifier” as an identifier that is generated depending on and being determined using “all of the data in the data item and only the data in the data item.” *See* ’791 at claims 1, 30, 33, & 35. The specification confirms this meaning by stating that a “substantially unique identifier” is an identifier determined using “all of the data in the data item[] and only on the data in the data item[.]” ’791 at 1:16-18; *see also id.* at 3:6-11. Where, as here, the inventor has acted as his own lexicographer, “the inventor’s lexicography governs.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed Cir. 2005). Thus, “substantially unique identifier” should be construed as “an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm.” This is the

very construction adopted by the District Court of Massachusetts in a prior action involving the '791 patent. *See Akamai Techs., Inc. & Massachusetts Inst. of Tech. v. Digital Island, Inc.*, 00-cv-11851, Dkt. No. 137 at 2 (attached as Exhibit 12).

Defendants offer no compelling reason to depart from the definition provided in the patents and adopted in another proceeding. Google's, Microsoft/Yahoo's, and NEC's proposed constructions each seek to include a requirement that the "substantially unique identifier" must be used only to "access" or for "accessing" a data item. But nothing in the specification limits the use of the identifier once it is generated. While some claims separately recite that "substantially unique identifiers" can be used to access data ('791 at claim 30), the claims and specification describe a variety of possible uses for a "substantially unique identifier" beyond "accessing" a "data item," such as using it to determine whether the data item already exists at a destination location and only transferring the data item to the destination location if it is not already there. *See, e.g.* '791 at claim 33. Thus, this Court should not add a "use" limitation into "substantially unique identifier"—particularly one use limitation when others are claimed.

Similarly, HP/Autonomy, Facebook, Google, and Microsoft/Yahoo all propose to include language purporting to state how "unique" the identifier must be (e.g., the identifier must be "substantially unique to [the] data" it is identifying or "virtually guaranteed to represent the data item and only the data item"). None of these proposed limitations follow the rules for claim construction. The claims and specification make clear that "substantially unique identifier" is an identifier determined in a particular way, not by how unique the identifier must be. Thus, a "substantially unique identifier" need not be "virtually guaranteed to represent ... only the data item." Indeed, the specification recognizes that at times even two different data items may end up having the same identifier value and that the function should be selected that tries to avoid

such “collisions.” ’791 at 13:10-67. In sum, what makes an identifier “substantially unique” is the use of a function that usually gives two different data items two different identifiers.

NetApp argues that the “term is not amenable to construction under 35 U.S.C. § 112.” NetApp is wrong. Claims are considered indefinite when they are “not amenable to construction or are insolubly ambiguous Thus, the definiteness of claim terms depends on whether those terms can be given any reasonable meaning.” *Young v. Lumenis, Inc.*, 492 F.3d 1336, 1346 (Fed. Cir. 2007) (quoting *Datamize, LLC v. Plumtree Software, Inc.*, 417 F.3d 1342, 1347 (Fed. Cir. 2005)). To make that determination, “in the face of an allegation of indefiniteness, general principles of claim construction apply.” *Young*, 492 F.3d at 1346.

Applying this law, the term “substantially unique identifier” is not insolubly ambiguous because it is expressly defined in both the claims and specification. Indeed, the specification gives a detailed description of a class of functions that may be used to produce a “substantially unique identifier,” along with examples of such functions (e.g., MD5 and SHA) that may be used to generate substantially unique identifiers, and explains how such functions may be used to generate a “substantially unique identifier” in one embodiment. *See* ’791 at 12:54-14:39 and Figs. 10(a)-10(b). Even NetApp provides a proposed construction in the alternative, thus contradicting its own indefiniteness argument.

Finally, NetApp’s proposed alternative construction is legally incorrect because it would limit the term “substantially unique identifier” only to “use in a computer network.” Neither the claim language nor the specification requires this limitation. The “substantially unique identifier” may be used in a computer network or even on a single computer. In fact, the specification expressly contemplates a data processing system embodiment that includes “one or more processors (or computers).” ’791 at 4:58-61 (emphasis added). Likewise, Figure 2 depicts

an exemplary file system of data items on a single processor. '791 at 5:29-34. Thus, limiting the claim term only to use in “a computer network” would violate the well-established rule against limiting the claims to a single embodiment by reading into claim language limitations from the specification. *Phillips*, 415 F.3d at 1323.

3. “Digital identifier” should be given its plain and ordinary meaning and construed as “an identifier in digital form.”

The term “digital identifier” underscores even further why construction of an “identifier” term either is unnecessary or, if construed, must be grounded in the claim language and specification. PersonalWeb does not believe that the Court needs to construe “digital identifier” separately as the term is defined by the claim itself. Specifically, the requirements for generating a digital identifier are precisely specified in claim 86 of the '310 patent. That claim provides:

A device operable in a network of computers, the device comprising hardware, including at least one processor and memory, to: (a) receive at said device, from another device in the network, a digital identifier for a particular sequence of bits, the digital identifier being based, at least in part, on a given function of at least some of the bits in the particular sequence of bits, wherein the given function comprises a message digest function or a hash function, and wherein two identical sequences of bits will have the same digital identifier

Thus, the term “digital identifier” at most can be construed to mean an identifier in digital form. Importantly, the words “at least some of the bits” make it plain that a digital identifier can be based on less than all of the data in a data item. Likewise, the words “at least in part” also make it plain that a digital identifier can be based on data or bits other than the data in the data item.

Ignoring the plain language of the claim, Defendants instead propose that “digital identifier” be construed to mean that the identifier is generated by “by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it

substantially unique to that data”⁴ Dkt. No. 157-2 at 14-15 (for example, “an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data item”).

The words of the claims themselves provide the starting point for any claim construction analysis. The Federal Circuit has instructed that “[i]n construing claims, the analytical focus must begin and remain centered on the language of the claims themselves, for it is that language that the patentee chose to use to ‘particularly point out and distinctly claim the subject matter which the patentee regards as his invention.’” *Interactive Gift Exp., Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1331 (Fed. Cir. 2001). Here, although the claim language makes clear that a digital identifier need only be based “at least in part, on a given function of at least some of the bits in the particular sequence of bits,” ’310 at claim 86 (emphasis added), Defendants nevertheless argue that a digital identifier must be generated by “processing all of the data in the data item and only the data in the data item.” Dkt. No. 157-2 at 14-15 (emphasis added). Defendants’ proposed construction is unsupported, and no reason exists for departing from the ordinary meaning of the claim language. *See Bell Atl. Network Servs., Inc.*, 262 F.3d at 1268. To the contrary, the separate use of the words “substantially unique” and “digital” in certain claims reflects a clear intent to use these different terms to refer to different processes in the generation of the identifier.

Nor does Defendants’ intrinsic and extrinsic evidence show why the term digital identifier should be construed as Defendants suggest. Defendants simply direct the Court to

⁴ While defendants each offer slightly different variations on the constructions, all share the common theme of importing the “all the data of the data item and only the data in the data item” concept which is inconsistent with the claim language. *See* Ex. 2; Dkt. No. 157-2 at 7-8, 11-32.

“[s]ee the intrinsic and extrinsic evidence cited for ‘substantially unique identifier.’” Dkt. No. 157-2 at 15. But “substantially unique identifier” is a different term from a different claim in a different patent, and there is no reason to assume that the intrinsic and extrinsic evidence supporting its construction should also apply to the term “digital identifier.” Thus, Defendants have no basis for the construction they propose.

4. “Data item identifier” should be given its plain and ordinary meaning and construed to mean “an identifier for a data item.”

The parties’ competing constructions as to the term “data item identifier” once again highlight Defendants’ incorrect attempts to graft a limitation onto the broad “identifier” terms that is contradicted by the claim language. PersonalWeb does not believe construction of “data item identifier” is needed. In the alternative, PersonalWeb proposes the Court construe this term as “an identifier for a data item.” Further explication is unwarranted because the claim language explains how this identifier is generated. Specifically, the requirements for generating a “data item identifier” are precisely specified in claim 81 of the ’096 patent. That claim provides:

A computer-implemented method operable in a file system comprising (i) a plurality of servers; (ii) a database; and (iii) at least one computer connected to the servers, the method comprising: obtaining, at said at least one computer, a first data item identifier for a first data item, said first data item consisting of a first plurality of non-overlapping segments, each of said segments consisting of a corresponding sequence of bits, and each of said segments being stored on multiple servers of said plurality of servers in the file system, said first data item identifier being based at least in part on the data comprising the first data item . . .

’096 at claim 81 (emphasis added). As with “digital identifier,” the words “at least in part” make it plain that a “data item identifier” can be based on data or bits other than all the data in the data item.

Notwithstanding this unambiguous claim language, Defendants take another run at arguing that this “identifier term” must be generated by “processing all of the data in the data

item and only the data in the data item.” Dkt. No. 157-2 at 7-8. Defendants, however, have no basis for asking the Court to ignore the ordinary meaning of the claim language and to instead adopt a construction that is contrary to the definition provided in the claim itself. *See Bell Atl. Network Servs., Inc.*, 262 F.3d at 1268.

Defendants here attempt again to conflate the meaning of “substantially unique identifier” and an unrelated term, this time “data item identifier.” These are two different species of “identifier.” A data item identifier may also be a “substantially unique identifier” if that data item identifier happens to be “determined using and depending on all of the data in the data item and only the data in the data item.” But “data item identifier”—a separate and distinct term—need not be “determined using and depending upon all of the data in the data item and only the data in the data item.” The express language of claim 81 makes clear that the data item identifier may be “based at least in part on the data comprising the data item,” in which case it would conclusively not meet the definition of “substantially unique identifier” (being “determined using and depending on all of the data in the data item and only the data in the data item”). *See* ’096 at claim 81.

Defendants notably omit any reference to the claim language in their proposed construction and instead urge the Court to “[s]ee the intrinsic and extrinsic evidence cited for ‘substantially unique identifier.’” Dkt. No. 157-2 at 15. Again, Defendants’ reliance on the intrinsic and extrinsic evidence of a different term from a different patent’s claims is inappropriate and provides no support for their impermissible construction.

5. “Data identifier” should be construed the same as “substantially unique identifier.”

The term “data identifier” is used in asserted claims 10, 15, 16 and 25 of the ’280 patent and claim 41 of ’791 patent. Unlike the other identifier terms discussed *supra*, PersonalWeb

agrees with Defendants that the Court should construe “data identifier” in the same way as it construes “substantially unique identifier.” Here, the specification is clear that data identifier has the same meaning as “substantially unique identifier.” “In the following, the terms ‘True Name’, ‘data identify’, and ‘data identifier’ refer to the substantially unique data identifier for a particular data item.” ’791 at 6:6-8. Thus, in direct contrast to “identifier” standing alone and with the other “identifier terms” discussed above, the specification expressly indicates that the patentee intended to equate the meaning of these related terms. The parties disagree, however, on the proper construction of “substantially unique identifier.” *See supra* at Section A.2.⁵

B. Claim Construction Analysis for Remaining Terms.

- 1. “True Name” should be construed as “the substantially unique identifier for a particular data item, calculated in accordance with the description at ’791 12:54 to 13:9.”**

The parties agree that “True Name” (a term coined by the inventors) refers to a substantially unique identifier that is calculated according to specific requirements set forth in detail in the specification. However, the parties disagree on the portions of the specification that apply.

⁵ The discussion above does not reflect the full extent of Defendants’ efforts to import the definition of “substantially unique identifier” to unrelated terms. Although Defendants agreed to limit the number of terms to be construed at this time based on this Court’s order, they have reserved their rights to raise nearly identical arguments concerning a slew of other “identifier terms.” *See* Dkt. No. 157 at 6 (“Defendants believe that all identifier-type terms (listed below) should be construed, but, in view of the Court’s order, have presented a subset of terms for immediate construction.”). These other “identifier terms,” which Defendants mostly argue should be construed to mean the same thing as their proposed construction of “substantially unique identifier,” include at least: “digital data item identifier”; “file identifiers”; “first identifier”; “digital part identifier”; “segment identifier”; “content-based identifier”/“content-dependent identifier”; and “substantially unique data identifier.” As explained in this brief, Defendants have no basis for asking the Court to ignore the ordinary meaning of these terms, which define the processes for generating the identifiers, and instead adopt a construction that is contrary to the unambiguous definition provided in the claim itself. *See Bell Atl. Network Servs., Inc.*, 262 F.3d at 1268.

a) The specification provides a clear description of how to calculate a True Name.

PersonalWeb's proposed construction for "True Name" relies on the specification, which clearly defines how a True Name is calculated:

1. Calculate True Name

A True Name is computed using a function, MD, which reduces a data block B of arbitrary length to a relatively small, fixed size identifier, the True Name of the data block such that the True Name of the data block is virtually guaranteed to represent the data block B and only data block B.

The function MD must have the following properties

'791 at 12:54-61 (emphasis added); *see also* '539 14:24-30 (same).⁶ The specification goes on to list five essential properties that the function must have. '791 at 12:62-13:9; '539 14:31-45 (same). This procedure defines the necessary parameters for calculating a True Name, and therefore correctly is referenced in PersonalWeb's construction.

b) Defendants' proposed construction improperly limits the claims to a preferred embodiment.

Defendants ignore the specification's description of the requirements for the general procedure for calculating a True Name and instead premise their claim construction contentions on a discussion of a specific mechanism for calculating a True Name within a preferred embodiment.

After first prescribing the necessary procedure to calculate a True Name, the specification addresses certain preferred embodiments. In particular, the specification discusses calculation of a True Name given a data item, but makes clear that this discussion involves only one preferred

⁶ Although PersonalWeb's construction references language from the '791 patent specification, this language is identical to that of the '539 specification. For clarity, PersonalWeb cites the corresponding language from both the '791 and '539 specifications through this section.

embodiment. *See* '791 at 55-60 (“While the invention is described herein using only the True Name of a data item as the identifier for the data item, other preferred embodiments use tagged, typed, categorized or classified data items and use a combination of both the True Name and the tag, type, category or class of the data item as an identifier.”); '539 at 15:24-29 (same). Defendants impermissibly rely on this description of a preferred embodiment for calculating a True Name for one species of identifier (a data item identifier) to limit the meaning of the claim language for the calculation of any True Name. This construction improperly limiting the claim to one preferred embodiment is contrary to well-established patent law and should be rejected.

2. “Data item(s)” should be construed as “sequence of bits.”

a) The specification defines “data item” as “sequence of bits.”

PersonalWeb’s proposed construction for “data item” is the definition provided by the specification:

In general, the terms “data” and “data item” as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits.

'791 at 1:54-60 (emphasis added); *see also id.* at 1:65-2:3. The Patentees expressly provided a non-limiting list of examples and made clear that any digital information represented by a sequence of bits qualifies as a “data item.” Where, as here, the inventor has acted as his own lexicographer, “the inventor’s lexicography governs.” *Phillips*, 415 F.3d at 1316. The patent language is clear that “data item” refers to “sequence of bits.” Therefore, PersonalWeb’s proposed construction should be adopted.

b) Defendants' proposed constructions are wholly unsupported by the intrinsic evidence.

Defendants' constructions ignore the Patentees' expressed "sequence of bits" definition of "data item" and seek instead to limit the construction of "data item" to the non-exclusive list of examples provided in the specification. But the specification lists these types of data items by way of example only, and Defendants' attempt to transmogrify non-exclusive examples into requirements ignores the language of the specification.

Neither the claim language nor the non-claim portion of the specification teach or suggest that data items must be limited to the identified examples. To the contrary, the patent lists these types of data items as examples of what a data item "may be." '791 at 1:54-60 ("Thus a data item may be the contents of a file.") (emphasis added). Indeed, the specification later provides other, different examples of what may constitute a data item, which are not included in Defendants' proposed construction. *Id.* at 1:65-2:3 ("In all of the prior data processing systems the names or identifiers provided to identify data items (the data items being files, directories, records in the database, objects in object-oriented programming, locations in memory or on a physical device, or the like) are always defined relative to a specific context."). Defendants' use of only some non-exclusive examples to limit the term is inconsistent with the specification.

Defendants Google, Microsoft, and Yahoo! take this manipulation of the specification's language a step further by rewriting the last example from "or other entity which can be represented by a sequence of bits" to "or other similar entity which can be represented by a sequence of bits." *See* Dkt. No. 157-2 at 2-3 (emphasis added). This attempt to further limit the term to a sequence of bits that is similar to the other stated examples lacks any support in the specification and claims. Thus, the Court should reject Defendants' attempt to import

nonexistent limitations and should instead adopt PersonalWeb's claim construction which follows the actual definitional language of the specification.

c) "Data item" does not render claim 30 ambiguous.

Microsoft and Yahoo! also argue that the "reference to 'data item' throughout claim 30 of the '791 patent is to the same data item; otherwise, the claim is indefinite." This is wrong. Claims are considered indefinite when they are "not amenable to construction or are insolubly ambiguous Thus, definiteness depends on whether claim terms can be given any reasonable meaning." *Young*, 492 F.3d at 1346. Here, data item is not ambiguous because the specification provides a clear definition and gives examples of numerous possible data items.

The reference to multiple "data items" in claim 30 of the '791 patent is not insolubly ambiguous. This claim provides:

A method of identifying a data item present in a data processing system for subsequent access to the data item, the method comprising:

determining a substantially unique identifier for the data item, the identifier depending on and being determined using all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier; and
accessing a data item in the system using the identifier of the data item.

'791 at claim 30. This claim expressly describes a system with multiple data items and recites that "two identical data items in the system will have the same identifier." *Id.* Further, the claim provides for "accessing a data item in the system using the identifier of the data item." *Id.* (emphasis added); see *Enzo Biochem, Inc. v. Applera Corp.*, 599 F.3d 1325, 1335 (Fed. Cir. 2010) (finding imprecise claim term not indefinite because the intrinsic evidence provided sufficient criteria for a person of ordinary skill to understand the scope of the claim). Following the explanation from the preceding provision, the identifier of a single data item may be used to

access any one of multiple instances of identical data items. This construction of the claim language gives the term a reasonable meaning and is not insolubly ambiguous. Accordingly, the claim is not indefinite.

3. “Data file(s)” should be construed as “a named data item having one or more data segments.”

a) PersonalWeb’s construction looks to the guidance provided in the specification.

PersonalWeb proposes that “data file” be construed as “a named data item having one or more data segments”—a construction that is consistent with the specification and its definitions.⁷

The True Name patent specification states:

A file is a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a single data segment. A compound file consists of a sequence of data segments. A data segment is a fixed sequence of bytes. An important property of any data segment is its size, the number of bytes in the sequence.

’791 at 5:45-54 (reference numerals removed).

Consistent with PersonalWeb’s proposed construction, the specification provides that a data file is “a named data item.” *Id.* at 5:47-48. As explained in the specification, that data file may be single or compound. *Id.* at 5:48-49. The specification provides further guidance that “a simple file consists of a single data segment” and that “a compound file consists of a sequence of

⁷ In its P.R. 4-3 filing, PersonalWeb proposed two alternative constructions for “data file.” In this brief, PersonalWeb clarifies its position and offers only a single proposed construction for “data file,” which is based entirely on the relevant language from the specification—namely, that “a simple file consists of a single data segment” and that “a compound file consists of a sequence of data segments.”

data segments.” The specification thus supports the clearest construction of “data file” as “a named data item having one or more data segments.”

b) Defendants’ proposed constructions import extraneous limitations into the term “data file.”

Defendants’ various constructions all import the limitation into the term “data file” that it must be in a directory. However, no definitional requirement in the specification for the term “data file” requires that all data files must be in directories. Defendants appear to rely upon the following portion of the specification to graft onto the term “data file” a requirement that it must be in a directory:

In other words, a file system 116 is a collection of directories 118.
A directory 118 is a collection of named files 120—both data files
120 and other directory files 118.

Id. at 5:45-47. But this portion of the specification is only directed to describing file systems and directories. It is not directed to defining the term “data file” and is inconsistent with the specification language that provides guidance on the term “data file” itself. Because Defendants’ constructions all import extraneous limitations into the term “data file,” they should be categorically rejected.

4. “File system” should have its ordinary meaning.

a) This term does not require construction.

“File system,” which appears in multiple claims, requires no construction. *See* ’662 claims 25, 30, 33; ’096 claims 1, 81, 83, 88, 97, 99. Rather, it should be given its plain and ordinary meaning. *See Altiris*, 318 F.3d at 1369 (acknowledging that there is a “heavy presumption” that claim terms carry their ordinary and customary meaning). Indeed, several of the Defendants appear to agree with PersonalWeb and have not asked the Court to construe this term. *See* Dkt. No. 157-2 at 12-13.

While PersonalWeb does not believe a construction for this term is required, if the Court determines it should be construed, PersonalWeb proposes that the term be construed to mean: “A collection of directories. A directory is a collection of named files.” This construction comes directly from the specification which states: “a file system is a collection of directories. A directory is a collection of named files.” ’791 at 5:44-49.

b) Defendants’ proposed constructions run afoul of the claim language.

Rather than use language from the specification defining “file system,” Defendants Google, Microsoft, Yahoo!, and NECAM manipulate the patent’s language to impose limitations on the claim term that do not exist. At least Defendants EMC and VMWare rely on actual language from the specification, though PersonalWeb maintains that this construction is unnecessary. Defendants Google, Microsoft, Yahoo!, and NECAM, by contrast, all construe “file system” to require that “each” directory “include” a collection of named files. *See* Dkt. No. 157-2 at 12-13. These constructions are simply not consistent with the specification, which defines a directory to be a collection of named files. In contrast, Defendants’ proposed construction makes “directory” and “collection of named files” two different things. Because Defendants’ construction separates directory into some undefined entity that “includes” a collection of files, rather than one that “is” a collection of files, it should be rejected.

5. “Sufficient number of copies” should be given its plain and ordinary meaning, or should be construed to mean “application-specific criteria regarding the number of copies required.”

The phrase “sufficient number of copies” has no special meaning beyond its plain and ordinary usage and requires no special construction. Indeed, several of the Defendants appear to agree with PersonalWeb and have either not asked the Court to construe this term or have agreed that the plain and ordinary meaning should apply. *See* Dkt. No. 157-2 at 13-14.

Alternatively, PersonalWeb proposes that the term be construed to mean “application-specific criteria regarding the number of copies required.” This construction is supported by the specification which provides: “This mechanism is used to ensure that files are available in alternate locations in mirror groups or archived on archival servers. The mechanism depends on application-specific migration/archival criteria (size, time since last access, number of copies required, number of existing alternative sources) which determine under what conditions a file should be moved.” ’791 at 26:22-28 (emphasis added). Although NECAM argues that this term should instead be construed to mean “[a] present number of copies greater than or equal to 2 of different locations in which a file is stored,” Dkt. No. 157-2 at 32, this construction is not supported by any of Defendants’ intrinsic evidence, Dkt. No. 157-2 at 32-33.

6. “Licensed/unlicensed” should be given its plain and ordinary meaning, or should be construed to mean “with license/without license.”

No construction is necessary for the terms “licensed” and “unlicensed” because these terms have a plain and ordinary meaning readily understood by a jury. *Phillips*, 415 F.3d at 1314.

Except for NEC, which agrees that the terms have a plain and ordinary meaning that requires no construction, each of the other Defendants assert that “licensed” (and “unlicensed”) mean that a person is “provided (or not provided) with a license to the content stored within the requested file.” What Defendants offer are not definitions for “licensed” and “unlicensed” because Defendants use the very terms they supposedly seek to construe in their proposed constructions. Instead, Defendants propose constructions that would limit the subject matter of a license—namely, to “content stored within a requested file.” Neither the claim language nor the specification limits licenses to “content stored within a requested file” or to any other content. For example, a “licensed” file can refer to simply “stored” content, irrespective of any request

for the content. As the specification explains, the substantially unique identifiers (True Names) can be used for active or passive enforcement of licensed content. With passive enforcement, the True Names are used to create a report of licensed users, regardless of whether any user has requested a file. '791 at 32:24-25. Similarly, the specification explains that a True Name can be recorded for each item in a license table, and then software can periodically compare the contents of a user processor with the license table. In this process, a user's status as "licensed" or "unlicensed" can be made without the user ever making a request for a file. '791 at 32:39-48. Thus, limiting the terms "licensed" and "unlicensed" as Defendants propose would improperly read out of the claim language entire embodiments described in the specification. *Phillips*, 415 F.3d at 1323. Defendants' constructions should be rejected.

7. "Distributing a set of data files across a network of servers" should be given its plain and ordinary meaning, or should be construed to mean "storing a set of data files on various servers in a network."

The phrase "distributing a set of data files across a network of servers" has an ordinary meaning that requires no construction. If the Court determines that this phrase requires construction, PersonalWeb proposes that it be construed as "storing a set of data files on various servers in a network." The specification and Figures 1(a) and 1(b) describe that within a system, data may be stored at various processor and various storage devices that are connected in some way, such as a network. '791 at 4:60-63. Thus, the meaning of "distributing a set of data files across a network of servers," when read in light of the specification, means "storing a set of data files on various servers in a network"

Defendants Apple, Microsoft, and Yahoo! propose a construction that would improperly limit the claim language to "delivering a set of data files to two or more servers." While the specification describes a multiprocessor system as one where more than one processor is used, there is nothing in the specification or claim language that limits the set of data files to be

“deliver[ed]... to two or more servers.” In fact, the specification contemplates a server/server peer-to-peer relationship. ’791 at 5:5-16. In such a configuration, there would only be two servers, and any data file that is “delivered” would be from a single server to another single server. Thus, limiting the claim language to “delivering a set of data files to two or more servers” would again read out of the claim language an entire embodiment described in the specification, and improperly read a new limitation into the claims. *Phillips*, 415 F.3d at 1323.

C. Construction Analysis for Means Plus Function Limitations.

1. “Access means for accessing a particular data item using the identifier of the data item.”

The parties agree that this limitation is recited in means-plus-function format with the function of “accessing a particular data item using the identifier of the data item.” The specification states that the “embodiment provided relies on the standard file management primitives for actually storing and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.” ’791 at 6:14-18. Thus, the corresponding structure to perform the recited function, as described in the specification, is “at least one processor (’791 at 7:62-63), programmed in accordance with one or more of the mechanisms described at ’791 33:51-34:32 (Make True File Local); 24:45-58; 17:10-45 (Request True File); 16:10-34 (Realize True File from Location); 21:52-66 (Open File); 25:25-38 (Acquire True File); and 35:38-41,51-61 (Locate Remote File).”

Defendants HP/Autonomy, EMC/VMware, and NEC all contend that the structure corresponding to the “access means” includes additional structure not tied to the function of “accessing a particular data item using the identifier of the data item.” See *Wenger Mfg. Inc. v. Coating Mach. Sys., Inc.*, 239 F.3d 1225, 1233 (Fed. Cir. 2001). Defendants seek to improperly import the “True File Registry” as part of the corresponding structure. But the True File

Registry does not perform the recited function; it is merely an input to the mechanisms that perform the recited function. *Id.* For this reason alone, Defendants' proposed constructions should be rejected.

Defendants also err by limiting the structure to a particular configuration comprised of "the processor of Fig. 1(b)" and "the 'Make True File Local' primitive mechanism depicted in Figures 17(a) and 17(b)." The specification lists many mechanisms, including primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms, that contain mechanisms for performing the recited function. *See* '791 at 6:19-7:36. Moreover, the specification notes that while it discloses the preferred embodiment(s), it is understood that the invention is intended to cover "various modifications and equivalent arrangements." '791 at 39:7-13. Thus, the specification does not require the particular configuration Defendants propose and should not be so limited. *See Callicrate v. Wadsworth Mfg.*, 427 F.3d 1361, 1369 (Fed. Cir. 2005) (holding that district court improperly interpreted "cutting means" to be restricted to the preferred "pivotally mounted cutting mechanism" instead of also including "scissors and hand-held razors" and other disclosed structures); *Versa Corp. v. Ag-Bag International Ltd.*, 392 F.3d 1325, 1329 (Fed. Cir. 2004) (reaffirming that "when multiple embodiments in the specification correspond to the claimed function, proper application of § 112, ¶ 6 generally reads the claim element to embrace each of those embodiments").

2. "Data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item."

The parties agree that this limitation is recited in means-plus-function format with the function of "making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item." The specification states that "the system provides transparent access to any data item by reference only to its identity and independent of its present

location.” ’791 at 3:54-56. It further states that the “embodiment provided relies on the standard file management primitives for actually storing and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.” ’791 at 6:14-18. Thus, the corresponding structure to perform the recited function, as described in the specification, is “at least one processor (’791 at 7:62-63), programmed in accordance with one or more of the mechanisms described at ’791 8:27-34; 9:36-37; 14:40 to 15:4; 35:46-48; 38:54-65.”

Defendants HP/Autonomy, EMC/VMware, and NEC all contend that the structure corresponding to the “data associating means” includes additional structure not tied to the function of “making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item.” *See Wenger*, 239 F.3d at 1233. Defendants again seek to include the “True File Registry” as part of the corresponding structure. But the True File Registry does not perform the recited function; it is merely an input to the mechanisms that perform the recited function. *Id.* Thus, Defendants’ proposed constructions should be rejected.

Defendants again err by limiting the structure to a particular configuration, comprised of “the processor of Fig. 1(b)” and “the ‘Assimilate Data Item’ primitive mechanism depicted in Figure 11.” The specification lists many mechanisms, including primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms, that contain mechanisms which perform the recited function. *See* ’791 at 6:19-7:36. Thus, the specification does not require the particular configuration Defendants propose, and again should not be limited as Defendants propose. *See Callicrate*, 427 F.3d at 1369; *Versa Corp.*, 392 F.3d at 1329.

3. “Existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.”

The parties agree that this limitation is recited in means-plus-function format with the function of “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” The specification states that the invention “further provides an apparatus and method for determining whether a particular data item is present in the system or at a location in the system by examining only the data identities of a plurality of data items.” ’791 at 3:35-39. One embodiment described in the specification “relies on the standard file management primitives for actually storing and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.” ’791 at 6:14-18. Thus, the structure described in the specification for performing the recited function is “at least one processor (’791 at 7:62-63), programmed in accordance with one or more of the mechanisms described at ’791 Fig. 14 (S260); Fig. 11 (S232); 15:54-57; 14:53-56 (beginning with “Next”); 16:25-45; 33:6-7; co. 8:27-32; Fig. 28; 23:53 to 24:13.” These are all routines that examine the identifiers of two or more data items to determine whether a particular data item is present in the system or at a location in the system.

HP/Autonomy and NEC argue that this element is indefinite under 35 U.S.C. § 112. Claims are considered indefinite when they are “not amenable to construction or are insolubly ambiguous Thus, the definiteness of claim terms depends on whether those terms can be given any reasonable meaning.” *Young*, 492 F.3d at 1346. The “existence means” limitation is not insolubly ambiguous. It is depicted in Figure 14 (S260) and Figure 11 (S232), and is described in, among other mechanisms, the “Locate True File” mechanism. (23:52-24:28). By offering an alternative construction, NEC also concedes that the limitation is capable of being given a reasonable meaning.

Defendants Amazon, EMC/VMware, NetApp, and NEC all contend that the structure corresponding to the “existence means” includes additional structure not tied to the function of “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” *See Wenger*, 239 F.3d at 1233. Like their proposed construction of the other means plus function limitations, Defendants seek to import the “True File Registry” as part of the structure. For the same reasons explained above, the True File Registry does not perform the recited function. *Id.*

Defendants further seek to limit the corresponding structure to a particular configuration comprised of “the processor of Fig. 1(b)” and “the ‘Locate Remote File’ primitive mechanism depicted in Figures 16(a) and 16(b).” Yet the specification lists many mechanisms that perform the recited function including primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms. *See* (6:19-7:36). Because the specification does not require the particular configuration Defendants propose, their construction should be rejected. *See Callicrate*, 427 F.3d at 1369; *Versa Corp.*, 392 F.3d at 1329.

4. **“Identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data on the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.”**

The parties disagree as to both the recited function and the corresponding structure for this means-plus-function limitation. Under the rules for construing terms governed by 35 U.S.C. § 112, the precise claim language in question must be examined to determine the appropriate means and function. “Correctly identifying the claimed function is critical, because ‘an error in identification of the function can improperly alter the identification of the structure . . . corresponding to that function.’” *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1087 (Fed. Cir. 2003) (internal citations omitted). PersonalWeb’s identification of the recited function is plain

from the claim language. It is “determining, for any of a plurality of data items present in the system, a substantially unique identifier.” *See Wenger*, 239 F.3d at 1233. Defendants seek to include within the function what the identity means uses to determine a substantially unique identifier (“all of the data in the data item and only the data in the data item”) and also the output of that determining—namely, “whereby two identical data items in the system will have the same identifier.” Because the results of “determining and identifier” are not a part of the determining process itself, “determining and identifier” should not be included as part of the recited function performed by the identity means.

As for the corresponding structure, the specification states, “[f]or any given data item, using the Calculate True Name primitive mechanism, a substantially unique identifier or True Name for that data item can be determined.” ’791 at 32:62-64. Thus, the structure for determining a substantially unique identifier is limited to “at least one processor (’791 at 7:62-63), programmed to perform the Calculate the True Name mechanism in accordance with ’791 12:54 to 13:19 and 14:1-39.”

Defendants all agree that the “Calculate True Name” mechanism determines a substantially unique identifier. Beyond that, Defendants incorrectly seek to limit the Calculate True Name mechanism to a particular configuration—namely, a function that produces “one of the MD4, MD5 and SHA functions.” But the specification makes clear that the Calculate True Name mechanism consists of a “family of functions” (see ’791 at 12:60-13:9) that “include MD4, MD5, and SHA.” (’791 at 13:10-14) (emphasis added). *See Callicrate*, 427 F.3d at 1369; *Versa Corp.*, 392 F.3d at 1329.

Finally, NetApp’s proposed construction would include additional structure that does not perform even the broadest reading of the function performed by the “identity means.” NetApp

proposes to include structure for determining whether the data item is simple or compound, as well as a branching decision tree based upon that determination. But none of these additional structures performs the function of determining a substantially unique identifier. Instead, these structures only make determinations of when to create a substantially unique identifier.

5. “Local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item.”

The parties’ disputes regarding the “local existence means” limitation are essentially the same as the “existence means” term discussed above. All parties except NEC agree that this means-plus-function limitation has the recited function of “determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items.” As for corresponding structure, the specification states that the invention “further provides an apparatus and method for determining whether a particular data item is present in the system or at a location in the system by examining only the data identities of a plurality of data items.” ’791 at 3:35-39. The specification further explains that the “embodiment provided relies on the standard file management primitives for actually storing and retrieving data items from disk, but uses the mechanisms of the present invention to reference and access those data items.” ’791 at 6:14-18. Thus, the structure described in the specification for performing the recited function is “at least one processor ’791 at 7:62-63, programmed in accordance with one or more of the mechanisms described at ’791 Fig. 14 (S260); ’791 at 15:54-57; 16:10-34; Fig. 28; 23:53 to 24:13.”

For the reasons discussed above, the arguments by HP/Autonomy and NEC that this limitation is indefinite under 35 U.S.C. § 112 lack merit. Likewise, the position asserted by Defendants Amazon, EMC/VMware, NetApp, and NEC that the structure corresponding to the “existence means” includes the “True File Registry” fails because the True File Registry does not perform the recited function, it is merely an input to the mechanisms that perform the recited

function. *Wenger*, 239 F.3d at 1233. Finally, Defendants' assertion that the corresponding structure limits the structure to a particular configuration (namely, "the processor of Fig. 1(b)" and "the 'Locate Remote File' primitive mechanism depicted in Figure 28") is legally incorrect because the specification lists many mechanisms that perform the recited function, including primitive mechanisms, operating system mechanisms, remote mechanisms, background mechanisms, and extended mechanisms, that contain mechanisms. See '791 at 6:19-7:36. *Callicrate*, 427 F.3d at 1369; *Versa Corp.*, 392 F.3d at 1329.

IV. CONCLUSION

In light of the foregoing, PersonalWeb respectfully requests that the Court adopt its claim constructions as outlined above.

DATED: June 5, 2013.

Respectfully submitted,

McKOOL SMITH, P.C.

/s/ Sam Baxter

Sam Baxter (Lead Counsel)
Texas State Bar No. 01938000
sbaxter@mckoolsmith.com
104 East Houston Street, Suite 300
Marshall, Texas 75670
Telephone: (903) 923-9000
Telecopier: (903) 923-9099

Theodore Stevenson, III
Texas State Bar No. 19196650
tstevenson@mckoolsmith.com
David Sochia
Texas State Bar No. 00797470
McKool Smith, P.C.
300 Crescent Court, Suite 1500
Dallas, Texas 75201
Telephone: (214) 978-4000
Telecopier: (214) 978-4044

Steve Pollinger
Texas State Bar No. 24011919
spollinger@mckoolsmith.com
McKool Smith, P.C.
300 West Sixth Street, Suite 1700
Austin, Texas 78701
Telephone: (512) 692-8700
Telecopier: (512) 692-8744

Roderick G. Dorman
rdorman@mckoolsmithhennigan.com
Lawrence M. Hadley
lhadley@mckoolsmithhennigan.com
McKool Smith, P.C.
865 South Figueroa Street, Suite 2900
Los Angeles, California 90017
Telephone (213) 694-1200
Telecopier: (213) 694-1234

Attorneys For Plaintiff
Personalweb Technologies, LLC

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing First Amended Complaint for Patent Infringement was filed electronically in compliance with Local Rule CV-5(a). As such, this document was served on all counsel who have consented to electronic service on June 5, 2013. Local Rule CV-5(a)(3)(A).

/s/ Lawrence M. Hadley

Lawrence M. Hadley

Exhibit 1

Exhibit 1

Agreed Constructions

Claim Term	Agreed Construction	Parties Agreeing to Construction
location(s) '791 Claims 2, 3, 33, 40 , 41 '280 Claim 54 '539 Claims 1, 21 '662 Claims 1, 4, 33 '096 Claims 1, 81, 83, 125	A particular processor in the system, a memory of a particular processor, a storage device, a removable storage medium, or any other physical location in the system.	PersonalWeb, Amazon, Apple, Microsoft, NECAM, NETAPP, and Yahoo!

Exhibit 2

Exhibit 2

PersonalWeb and Defendants’ Proposed Constructions

Index of Terms

1. Identifier.....Page 2

2. Substantially Unique IdentifierPages 3-4

3. Digital Identifier..... Page 5-6

4. Data Item Identifier.....Page 7

5. Data IdentifierPage 8

6. True NamePage 9

7. Data Item.....Pages 10-11

8. Data File(s).....Page 12

9. File System.....Page 13

10. Sufficient Number of CopiesPage 14

11. Licensed/UnlicensedPage 15

12. Distributing a Set of Data Files Across a Network of ServersPage 16

13. Access Means for Accessing a Particular Data Item Using the Identifier of the Data Item..
.....Page 17

14. Data Associating Means for Making and MaintainingPage 18

15. Existence Means for Determining Whether a Particular Data item is Present in the
SystemPages 19-20

16. Identity Means for Determining.....Pages 21-22

17. Local Existence Means for Determining Whether an Instance of a Particular Data Item is
Present at a Particular Location in the System.....Pages 23-24

1. Identifier

Disputed Term	PersonalWeb	NECAM
Identifier '791 Claims 1, 2,3, 4, 30, 31, 32, 33, 35, 41 '310 claims 1, 5, 6 '310 Claims 70, 81, 86 '539 Claims 10, 21, 34 '662 Claims 1, 3, 30 '096 Claims 1, 2, 81, 83	No construction necessary. Plain and ordinary meaning or defined by the claim itself. Alternatively, a value that identifies.	Plain and ordinary meaning
		NETAPP
		A name of a data item for use in a computer network
		Google
		an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data.

2. Substantially Unique Identifier

Disputed Term	PersonalWeb	HP & Autonomy
substantially unique identifier '791 Claims 1, 2,3, 4, 30, 31, 32, 33, 34, 35, 41	an identity for a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm. Alternatively, plain and ordinary meaning.	an identifier for access to a data item, generated by processing all of the data in the data item and only the data in the data item through an algorithm that makes the identifier substantially unique to those data
		EMC & VMWare & Apple
		an identity for a data item generated by processing <i>all</i> of the data in the data item, and <i>only</i> the data in the data item, through an algorithm
		Facebook
		an identifier, generated by processing all the data in the data item and only the data in the data item through an algorithm that makes the identifier substantially unique to those data.
		Google
		('791 claims 30-32, 35, 41, and "identifier" in dependent claims): an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data
		Microsoft and Yahoo!
		an identifier used to access a data item, generated by processing all the data in the data item and only the data in the data item through an algorithm that makes the identifier virtually guaranteed to represent the data item and only the data item.

		NECAM
		The name for accessing or referencing a data item in a system, generated by processing all of the data in the data item, and only the data in the data item, through an algorithm
		NETAPP
		This term is not amendable to construction under 35 U.S.C. § 112. In the alternative, to the extent this term means anything, it means: “a name for a data item for use in a computer network such that the data item can be definitively determined to be the same as or different from another data item in the computer network based solely on the name”

3. Digital Identifier

Disputed Term	PersonalWeb	HP & Autonomy
digital identifier '310 Claim 86	No construction necessary. Plain and ordinary meaning or defined by the claim itself. Alternatively, identifier in digital form.	an identifier for access to a data item that consists of a particular sequence of bits of data, the identifier being generated by processing all of the data in the data item and only the data in the data item through an algorithm that makes the identifier substantially unique to those data
		Apple
		an identifier for a particular sequence of bits generated at least in part by processing all of the bits in the sequence of bits, and only the bits in the sequence of bits, through an algorithm
		Facebook
		an identifier, generated by processing all the data in the data item and only the data in the data item through an algorithm that makes the identifier substantially unique to those data
		Google
		an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data
		Microsoft & Yahoo!
		an identifier used to access a data item, generated by processing all the data in the data item and only the data in the data item through an algorithm that makes the identifier virtually guaranteed to represent the data item and only the data item.

		NECAM
		The name for accessing or referencing a data item in a system, generated by processing all of the data in the data item, and only the data in the data item, through an algorithm

4. Data Item Identifier

Disputed Term	PersonalWeb	HP & Autonomy
data item identifier '539 Claim 34 '096 Claim 81	No construction necessary. Plain and ordinary meaning or defined by the claim itself. Alternatively, an identifier for a data item.	an identifier for access to a data item, generated by processing all of the data in the data item and only the data in the data item through an algorithm that makes the identifier substantially unique to those data
		NECAM
		The name for accessing or referencing a data item in a system, generated by processing all of the data in the data item, and only the data in the data item, through an algorithm

5. Data Identifier

Disputed Term	PersonalWeb	Apple
data identifier '280 Claims 10, 15, 16, 18, 25 '791 claims 33, 34, 41	the substantially unique identifier	an identifier for a particular data item generated at least in part by processing all of the data in the data item, and only the data in the data item, through an algorithm
		Google
		an identifier for access to a data item generated by processing all of the data in the data item, and only the data in the data item, through an algorithm that makes it substantially unique to that data
		Microsoft & Yahoo!
		An identifier used to access a data item, generated by processing all the data in the data item and only the data in the data item through an algorithm that makes the identifier virtually guaranteed to represent the data item and only the data item.
		Facebook
		an identifier, generated by processing all the data in the data file and only the data in the data file through an algorithm that makes the identifier substantially unique to those data

6. True Name

Disputed Term	PersonalWeb	EMC & VMware
True Name '539 Claim 34	The substantially unique identifier for a particular data item, calculated in accordance with the description at 12:54 to 13:9. ('791 patent)	substantially unique identifier for a particular data item based on all of the data in the data item and only the data in the data item, calculated in accordance with the description at 15, l. 37-col. 16, l. 8 ('539 patent)
		NECAM
		The substantially unique data identifier for a particular data item, calculated in accordance with the description at col. 15, l. 37 - 16, l. 8 of the '539 patent

7. Data Item(s)

Disputed Term	PersonalWeb	HP & Autonomy
data item(s) '310 Claims 1, 2, 7, 8, 10, 11, 14, 16-18, 19, 24, 32, 70, 81, 86 '791 Claims 1, 2, 3, 4, 29, 30, 31, 32, 33, 34, 35, 41 '662 Claims 1, 25, 30, 33 '096 Claim 1, 2, 81, 83, 86, 88, 125 '539 Claim 10, 21, 34 '544 Claims 1, 8, 13, 16, 19, 21 '420, Claim 166	sequence of bits	the contents of: a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits
		Facebook & NETAPP
		No construction necessary / plain and ordinary meaning
		Google
		No construction. <i>Alternatively</i> , "the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other similar entity which can be represented by a sequence of bits"
		Microsoft & Yahoo!
		The contents of: a file; a portion of a file; a page in memory; an object in an object-oriented program; a digital message; a digital scanned image; a part of a video or audio signal; or any other similar entity that can be represented by a sequence of bits. The reference to "data item" throughout claim 30 of the '791 patent is to the same data item; otherwise, the claim is indefinite.

		NECAM
		The contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits
		EMC & VMware
		A sequence of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits.

8. Data File(s)

Disputed Term	PersonalWeb	HP & Autonomy
data file(s) '280 Claims 10, 15, 16, 18, 25, 31, 32, 36. 38 '442 Claims 1, 2, 4, 7, 23, 30	a named data item having one or more data segments	a data item that is named in a directory of a file system
		EMC & VMware
		The patents state that a "file" is "a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a single data segment. A compound file consists of a sequence of data segments. A data segment is a fixed sequence of bytes" ('791 patent at 5:47-52). EMC and VMware are agreeable to a construction similar to PersonalWeb's proposed construction for "data file" in its recent filing with the U.S. Patent and Trademark Office, which Defendants believe is consistent with the definition in the asserted patents: "A named data item that appears in a directory in a file system and which is a data file (which may be simple or compound)."
		Facebook
		a named data item that appears in a directory
		Google
		No construction. <i>Alternatively</i> , "a named data item that appears in a directory in a file system and which is a data file"
		Microsoft & Yahoo! & Amazon
		A data item that is listed by its file name in a directory in a file system.

9. File System

Disputed Term	PersonalWeb	EMC & VMware
file system '662 Claim 25, 30, 33 '096 Claims 1, 81, 83, 88, 97, 99	No construction necessary. Plain and ordinary meaning or defined by the claim itself. Alternatively, a collection of directories. A directory is a collection of named files.	a collection of directories. A directory is a collection of named files – both data files and other directory files. A file is a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a single data segment. A compound file consists of a sequence of data segments
		Google
		“a collection of directories, which each include a collection of named files”
		Microsoft & Yahoo
		A collection of directories, which each include a collection of named data files or named directories.
		NECAM
		a collection of directories, which each include a collection of named files

10. Sufficient Number of Copies

Disputed Term	PersonalWeb	NECAM
sufficient number of copies	No construction necessary. Plain and ordinary meaning or defined by the claim itself.	A preset number of copies greater than or equal to 2 of different locations in which a file is stored
'662 Claim 1	Alternatively, application-specific criteria regarding the number of copies required	

11. Licensed/Unlicensed

Disputed Term	PersonalWeb	HP & Autonomy
licensed/unlicensed '442 Claim 1	No construction necessary. Plain and ordinary meaning or defined by the claim itself. Alternatively, with license/without license	for "licensed": "provided with a license to the content stored within the requested file"
		Amazon
		Licensed – "having a license to content stored within a requested file" Unlicensed - "not having a license to content stored within a requested file"
		NECAM
		Plain and ordinary meaning.
		EMC & VMware
		licensed: " provided with a license to content stored within the requested file" unlicensed: "not provided with a license to content stored within a requested file"

12. Distributing A Set of Data Files Across A Network of Servers

Disputed Term	PersonalWeb	Apple / Microsoft / Yahoo!
Distributing a set of data files across a network of servers	No construction necessary. Plain and ordinary meaning or defined by the claim itself.	Delivering a set of data files to two or more servers in a network of servers.
'280 Claims 10, 15, 16, 25, 31	Alternatively, storing a set of data files on various servers in a network.	

13. Access Means for Accessing a Particular Data Item Using the Identifier of the Data Item.

Disputed Term	PersonalWeb	HP & Autonomy
<p>access means for accessing a particular data item using the identifier of the data item</p> <p>'791 Claim 4</p>	<p>Function: Accessing a particular data item using the identifier of the data item. [Agreed]</p> <p>Structure: At least one processor (7:62-63), programmed in accordance with one or more of the mechanisms described at '791 33:51 to 34:32; 24:45-58; 17:10-45; 16:10-34; 21 :52-66; 25:25-38; 35:38-41,51-61.</p>	<p>Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Make True File Local” primitive mechanism depicted in Figures 17(a) and 17(b). '791 patent at 17:10-45.</p>
		EMC & VMware
		<p>Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Make True File Local” primitive mechanism depicted in Figures 17(a) and 17(b). <i>See</i> '791 patent at 4:64-6:19, 9:36-10:10, 17:10-45, 18:4-8, 18:53-55, 20:65-21:5, 24:54-55, 33:51-59, 36:52-55, 36:61-64.</p>
		NECAM
		<p>Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Make True File Local” primitive mechanism depicted in Figures 17(a) and 17(b).</p>

14. Data Associating Means for Making and Maintaining

Disputed Term	PersonalWeb	HP & Autonomy
data associating means for making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item '791 Claim 4	Function: Making and maintaining, for a data item in the system, an association between the data item and the identifier of the data item. [Agreed] Structure: At least one processor (7:62-63), programmed in accordance with one or more of the mechanisms described at '791 8:27-34; 9:36-37; 14:40 to 15:4; 35:46-48; 38:54-65.	Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Assimilate Data Item" primitive mechanism depicted in Figure 11 (S232, S236, S237, S238, S239). '791 patent at 14:53-15:4.
		EMC & VMware
		Structure The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Assimilate Data Item" primitive mechanism depicted in Figure 11. <i>See</i> '791 patent at 4:64-6:19, 9:36-10:10, 14:40-15:4, 15:41-44, 16:29-31, 18:34-36, 18:43-45, 19:30-37, 24:34-35, 24:51-52, 28:30-33, 30:55-57, 32:54-33:9, 33:33-39.
		NECAM
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Assimilate Data Item" primitive mechanism depicted in Figure 11.

15. Existence Means for Determining Whether a Particular Data item is Present in the System

Disputed Term	PersonalWeb	Amazon
existence means for determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items '791 Claim 1	<p>Function: Determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items. [Agreed]</p> <p>Structure: At least one processor (7:62-63), programmed in accordance with one or more of the mechanisms described at '791 Fig. 14 (S260); Fig. 11 (S232); 15:54-57; 14:53-56 (beginning with "Next"); 16:25-45; 33:6-7; co. 8:27-32; Fig. 28; 23:53 to 24:13.</p>	<p>Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate Remote File" primitive mechanism depicted in Figures 16(a) and 16(b). See '791 patent at 4:64-6:19, 9:36-10:10, 16:38-17:9, 17:41-43, 25:10-13, 35:51-55.</p>
		HP & Autonomy
		Structure: This element is invalid as indefinite under 35 U.S.C. § 112.
		NECAM
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate Remote File" primitive mechanism depicted in Figures 16(a) and 16(b).
		NetApp
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the mechanism depicted in step S232 of Figure 11 and the "Locate Remote File" primitive mechanism depicted in Figures 16(a) and 16(b).

		EMC & VMware
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Locate Remote File” primitive mechanism depicted in Figures 16(a) and 16(b).

16. Identity Means for Determining

Disputed Term	PersonalWeb	Amazon
<p>identity means for determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier</p> <p>'791 Claim 1</p>	<p>Function: Determining, for any of a plurality of data items present in the system, a substantially unique identifier</p> <p>Structure: At least one processor (7:62-63), programmed to perform the Calculate the True Name mechanism in accordance with '791 12:54 to 13:19 and 14:1-39</p>	<p>Function: Determining, for any of a plurality of data items present in the system, a substantially unique identifier, the identifier being determined using and depending on all of the data in the data item and only the data in the data item, whereby two identical data items in the system will have the same identifier.</p> <p>[Agreed by all Defendants listed]</p> <p><i>Amazon:</i></p> <p>Structure: The processor of Fig. 1(b), programmed to execute the "Calculate True Name" primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions.</p>
		HP & Autonomy
		<p>Structure: The processor of Fig. 1(b), programmed to execute the "Calculate True Name" primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions. '791 patent at 12:54-13:19, 14:1-35.</p>
		EMC & VMware
		<p>Structure: The processor of Fig. 1(b), programmed to execute the "Calculate True Name" primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions. See '791 patent at 4:64-6:19, 12:54-14:39, 14:51-53, 31:32-50, 32:54-64.</p>

		NECAM
		Structure: The processor of Fig. 1(b), programmed to execute the “Calculate True Name” primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions.
		NetApp
		Structure: The processor of Fig. 1(b), programmed to execute the “Calculate True Name” primitive mechanism depicted in Figures 10(a) and 10(b), where the MD function is one of the MD4, MD5 and SHA functions, including determining if the data item is a simple or compound data item, and when it is a compound data item then applying a MD4, MD5 or SHA has function to each of the segments of the compound data item to obtain a True Name for each of the segments, and creating a block consisting of the respective True Names of the plurality of segments, and then applying the hash function performed in computing a True Name to the block of segment True Names in order to compute the True Name of the compound data item.

17. Local Existence Means for Determining Whether an Instance of a Particular Data Item is Present at a Particular Location in the System

Disputed Term	PersonalWeb	Amazon
local existence means for determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item. '791 Claims 2, 3	Function: Determining whether a particular data item is present in the system, by examining the identifiers of the plurality of data items. [Agreed by all Defendants listed except NECAM] Structure: At least one processor (7:62-63), programmed in accordance with one or more of the mechanisms described at '791 Fig. 14 (S260); 15:54-57; 16:10-34; Fig. 28; 23:53 to 24:13.	Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate Remote File" primitive mechanism depicted in Figures 16(a) and 16(b)
		HP & Autonomy
		Structure: This element is invalid as indefinite under 35 U.S.C. § 112.
		EMC & VMware
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate True File" remote mechanism depicted in Figure 28.
		NECAM
		This element is invalid as indefinite under 35 U.S.C. § 112. In the alternative: Function: Determining whether an instance of a particular data item is present at a particular location in the system, based on the identifier of the data item. Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the "Locate True File" remote mechanism depicted in Figure 28.

		NetApp
		Structure: The processor of Fig. 1(b), storing the True File Registry and programmed to execute the “Locate True File” remote mechanism depicted in Figure 28.